Interviews

The Process

The process can be: a referral or clearing a Resume screen, test cases on a coding round, a code review and second Resume screen, interviews, and negotiating [optionality, one can precompute a monetary threshold from a firm such as 550000/550000/550000 for signing/salary/bonus]. Practice interviewing and obtain offers to use with other firms. Save and review all .txt file interviews notes and coding rounds. Prepare notes, questions to ask, stories to tell, private research ideas, maximal literature and research referentiality, and execute strong robust post facto analysis, editing, and sharpening. Smile, be pleasant, and bring up the interviewers' first names, LinkedIns, and public written records like "impressive courses [X]". On applications forms, check "I don't wish to answer" except perhaps "Yes" for disability.

Resume Writing

Minimise clutter; most people know that the University Of Texas At Austin is in Austin, Texas. "Putnam - 93rd" rather than "Honourable Mention" because the reader might care but not know what something means in terms of rank. "C++ [>25000 Lines]". Do not represent anything less than intermediate skill, and back up staked claims of knowledge. If one writes "NumPy", skim a topical book. Or, at the very least, use [ctrl a] [ctrl c] [ctrl v] to merge 10 "cheat sheets" and compilations into a Python.py file. Consider website links. Dates for degrees. Make experience sound impressive. Write down most things someone might care about so this includes "Maths [Age 12], Reading, Maths 2, Physics SAT 800s, 10 AP 5s". Use the .pdf file format: "Resume - Lazar Ilic.pdf". Either via a Cover

Letter or public GitHub README/LinkedIn biography, represent “passion” [implicitly for work]. People may not observe one’s consistent commit history, or codes and content production. One can draw attention to these.

Interviews

Solve through Heard On The Street “Volumes 1-100”. Be clear and technical, audible via microphone [volume regularisation function]. Consider lighting, camera, microphone, image brightening functions, background, etc. Consider the video option, if it exists, over the dial in phone option. One can use a custom Black background in the Zoom application and perhaps video call from bed.

The point of a technical interview is to be technical. If one thinks the Examples below are too detail oriented, think again. Clarity is valued, aspire to being sharper than extant written solutions. These would have one thinking that something is adequate when it is not, depending on the interviewer. In trading, people effectively communicate. Aspire to implement tasks in 1 minute with 0 errors in a live shared coding editor whilst nonchalantly bantering about market movements and the news.

Firms are free to ask basically whatever: open maths problems, tricky puzzles, to explain the most basic textbook ideas and theorems. They will likely not expect anything too deep or obscure, so understand key ideas like those one finds in interview books. Be prepared, calm, try low n cases, make true observations out loud, and consider brainstorming ideas out loud too. This will let one’s interviewer know that one has ideas. The more relevant maths words said out loud, the better. Correct answers and sound logical reasoning are minor steps on the way towards nailing an interview. Be ready to talk about anything on one’s Resume and prepare a narrative about recent activity.

They will only have a few hours of exposure. Make these hours count. This is about a group in a competitive domain, so impress them, but also try and signal that one will be a good colleague. Have well composed text files open to read and the internet to query. Regularly review a sheet of formulae and have open during rounds. When drafting solutions, gesture towards general notions or cases to show that one understands the broader structures in which these tasks are embedded.

Clear browser cookies on GlassDoor.com and scrape all questions from all firms into a text file to fully solve. Practicing the composition of explanations helps one become performant in interviews.

If one instantly replies with a precomposed answer and they ask if one has seen the task before, say “yes”, otherwise do not mention it except if it is extremely canonical where bringing this up is plausibly worth doing to demonstrate knowledge of the “canon”. If one cannot remember a formula, do not say “this is canonical”. Rather, say something like “I could look this up in my .pdf of Theory Of Probability by Gordan Zitkovic”. Or say one is good at searching the literature, Stack, ArXiV, Google Scholar, SciHub.

Coding Rounds

Use reasonable complete English words, imitate capitalisation style, and indicators like “a” as variable names. Abbreviations and shortened words are extremely ambiguous and constitute very bad writing. Write robust code. This should be precomputed and memorised.

Write ≈ 10 lines of smart comments per task with key ideas about algorithms, time and memory asymptotics, optimality, edge cases, low n cases, input limits, etc. Write a 1 line return statement program which clears 8/10 test cases if one exists. Avoid axiomatics, models of computation, and theoretical computer science style asymptotic analyses in bits of input. Clarify the supposition that all values are bounded floats so that many operations are treated as $O(1)$.

24-168 **Hour Coding Rounds**

Schedule at least 7 sessions of 3 hours to work and send on the due date. Ensure both the code and lengthy writing are legible. Include ideation, reference texts used, and one’s process.

Emails Writing

Optimise to maximise expected value, obtaining and signing desired offer.

Hi Ms./Mr. X,

Thanks! I look forward to meeting Y.

[Consider follow up comments on the previous round, generalising tasks e.g.]

Sincerely,

Z

Probability And Statistics

Random Variable X	Discrete	Continuous
Cumulative Distribution Function	$F(a) = P\{X \leq a\}$	$F(a) = \int_{-\infty}^{a} f(x)dx$
Probability Mass/Density Function	$p(x) = P\{X = x\}$	$f(x) = \frac{d}{dx}F(x)$
Expected Value	$\sum p(x) \cdot x$	$\int_{-\infty}^{\infty} f(x) \cdot x dx$
Expected Value Of $g(x)$	$\sum p(x) \cdot g(x)$	$\int_{-\infty}^{\infty} f(x) \cdot g(x) dx$

	Probability Mass Function	E$[X]$	Var(X)
Uniform	$\frac{1}{b-a+1}, x \in [a, b]$	$\frac{b+a}{2}$	$\frac{(b-a+1)^2}{12}$
Binomial	$\binom{n}{x} p^x (1-p)^{n-x}, x \in [0, n]$	np	$np(1-p) = npq$
Poisson	$\frac{e^{-\lambda t}(\lambda t)^x}{x!}, x \in [0, \infty]$	λt	λt
Geometric	$(1-p)^{x-1}p, x \in [1, \infty]$	$\frac{1}{p}$	$\frac{1-p}{p^2} = \frac{q}{p^2}$
Negative Binomial	$\binom{x-1}{r-1} p^r (1-p)^{x-r}, x \in [r, \infty]$	$\frac{r}{p}$	$\frac{r(1-p)}{p^2} = \frac{rq}{p^2}$

	Probability Density Function	E$[X]$	Var(X)
Uniform	$\frac{1}{b-a}, x \in [a, b]$	$\frac{b+a}{2}$	$\frac{(b-a)^2}{12}$
Normal	$\frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{-(x-\mu)^2}{2\sigma^2}}, x \in [-\infty, \infty]$	μ	σ^2
Exponential	$\lambda e^{-\lambda x}, x \in [0, \infty]$	$\frac{1}{\lambda}$	$\frac{1}{\lambda^2}$
Gamma	$\frac{\lambda e^{-\lambda x}(\lambda x)^{\alpha-1}}{\Gamma(\alpha)}, x \in [0, \infty], \Gamma(a) = \int_0^\infty e^{-y} y^{a-1}$	$\frac{\alpha}{\lambda}$	$\frac{\alpha}{\lambda^2}$
Beta	$\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}, x \in (0, 1)$	$\frac{\alpha}{\alpha+\beta}$	$\frac{\alpha\beta}{(\alpha+\beta+1)(\alpha+\beta)^2}$

My “Probability Models” and “Financial Mathematics For Actuarial Applications” corpi contain much more content.

Variance Of X Var(X):
$\mathrm{E}[(X-\mathrm{E}[X])^2]=\mathrm{E}[X^2]-(\mathrm{E}[X])^2$

Standard Deviation Of X Sd(X):
$\sqrt{\mathrm{Var}(X)}$

Covariance:
$\mathrm{Cov}(X,Y)=\mathrm{E}[(X-\mathrm{E}[X])(Y-\mathrm{E}[Y])]=\mathrm{E}[XY]-\mathrm{E}[X]\mathrm{E}[Y]$

Covariance Matrix Is
Positive Semidefinite

$$\begin{bmatrix} \mathrm{Var}(X_1) & \mathrm{Cov}(X_1,X_2) & \dots \\ \mathrm{Cov}(X_1,X_2) & \mathrm{Var}(X_2) & \dots \\ \dots & \dots & \dots \end{bmatrix}$$

Correlation:
$\mathrm{Corr}(X,Y)=\rho_{X,Y}=\frac{\mathrm{Cov}(X,Y)}{\sqrt{\mathrm{Var}(X)\mathrm{Var}(Y)}}$

Correlation Matrix Is
Positive Semidefinite

$$\begin{bmatrix} 1 & \rho_{X_1,X_2} & \dots \\ \rho_{X_1,X_2} & 1 & \dots \\ \dots & \dots & \dots \end{bmatrix}$$

Bernoulli Distribution:
$x\in[0,1]$
$[1-p,p]$
$\mathrm{E}[X]=p$
$\mathrm{Var}[X]=p(1-p)$

Bernoulli Fair Coin Flip Bet:
$x\in[-1,1]$
$\left[\frac{1}{2},\frac{1}{2}\right]$
$\mathrm{E}[X]=0$
$\mathrm{Var}[X]=\text{Std Dev}[X]=1$

Laplace Rule Of Succession:
s Successes In n Binary Observations
Posterior From Uniform Prior $p\in[0,1]$
$\hat{p}=\frac{s+1}{n+2}$

Binomial Distribution (n,p):
Sum Of Bernoulli Variables e.g.
$x\in[0,1,\dots,n]$
$[\binom{n}{0}p^0(1-p)^n,\dots]$
$\mathrm{E}[X]=np$
$\mathrm{Var}[X]=np(1-p)=npq$
$\mathrm{Var}[\mathrm{BinomialProportion}(n,p)]=\frac{p(1-p)}{n}=\frac{pq}{n}$

Geometric Distribution:
$x\in[0,1,2,\dots]$
$[p,(1-p)p,(1-p)^2p,\dots]$
$\mathrm{E}[X]=\frac{1-p}{p}$
$\mathrm{Var}[X]=\frac{1-p}{p^2}$

Poisson Distribution:
$x\in[0,1,2,\dots]$
$[e^{-\lambda}\frac{\lambda^x}{x!}]$
$\mathrm{E}[X]=\lambda$
$\mathrm{Var}[X]=\lambda$

Uniform Distribution:
$x\in[a,b]$
$f(x)=\frac{1}{b-a}$
$\mathrm{E}[X]=\frac{a+b}{2}$
$\mathrm{Var}[X]=\frac{(b-a)^2}{12}$

Normal/Gaussian Distribution
$X\sim N(\mu,\sigma)$:
$x\in[-\infty,\infty]$
$f(x)=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$
$\mathrm{E}[X]=\mu$
$\mathrm{Var}[X]=\sigma^2$

Exponential Distribution $\tau>0$ Mean

Parametrisation:
$x \in [0, \infty]$ $\;f(x) = \frac{1}{\tau} e^{-\frac{x}{\tau}}$
$\mathrm{E}[X] = \tau$
$\mathrm{Var}[X] = \tau^2$

$\chi^2(n)$ Distribution:
Sum Of n Squared $N(0,1)$s
$x \in [0, \infty]$
$f(x) = \frac{1}{2^{\frac{n}{2}} \Gamma(\frac{n}{2})} x^{\frac{n}{2}-1} e^{-\frac{x}{2}}$
$\mathrm{E}[X] = n$
$\mathrm{Var}[X] = 2n$

$\Gamma(k, \tau)$ Gamma Distribution:
$x \in [0, \infty]$
$f(x) = \frac{1}{\Gamma(k)\tau^k} x^{k-1} e^{-\frac{x}{\tau}}$
$\mathrm{E}[X] = k\tau$
$\mathrm{Var}[X] = k\tau^2$

Power Law Distribution $\forall a > 3$:
$x \in [1, \infty]$
$f(x) = \frac{1}{(a-1)x^a}$
$\mathrm{E}[X] = \frac{1}{a^2-3a+2}$
$\mathrm{Var}[X] = \frac{1}{a^2-4a+3} - \left(\frac{1}{a^2-3a+2}\right)^2 = \frac{a^3-5a^2+7a-1}{(a-3)(a-2)^2(a-1)^2}$

More heavy-tailed than Log Normal Distributions. Linear on a log-log plot [plotting both the x and y-axes on log scales]. Examples include words' multiplicities in a TV scripts corpus, US "city" populations, Twitter followers counts over all users. Scale-invariant. One reason is rich get richer phenomena.

Log Normal Distribution:
$f(x) = \mathrm{Lognormal}(\mu, \sigma^2) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{(\ln(x)-\mu)^2}{2\sigma^2}}$
$\mathrm{E}[X] = e^{\mu + \frac{\sigma^2}{2}}$
$\mathrm{Var}[X] = (e^{\sigma^2} - 1)e^{2\mu+\sigma^2}$

Multiplicative factors tend to occur whenever there is a "growth" process over time. Much more heavy-tailed than normal distributions. Example maybe stock prices of SP 500 stocks.

Inverse Exponential Distribution:
$x \in [0, \infty]$
$f(x) = \frac{1}{tx^2} e^{-\frac{1}{tx}}$
$F(x) = e^{-\frac{1}{tx}}$

In these interview approximation tasks maybe one thinks that the logarithm of one's estimate is roughly normal so when one multiplies there is a reduction in relative variance in the exponent term. Perhaps a strategy is to execute such a task with single value estimators and then when asked for a credence interval throw out something like $\left[\frac{1}{5} \cdot x, 5x\right]$.

If X and Y are independent,
$\mathrm{Cov}(X, Y) = 0$ and
$\mathrm{Corr}(X, Y) = \rho_{X,Y} = 0$

$\mathrm{Cov}\left(\sum a_i X_i, b_j Y_j\right) = \sum a_i b_j \mathrm{Cov}(X_i, Y_j)$

$\mathrm{Var}\left(\sum X_i\right) = \sum \mathrm{Var}(X_i) + 2\sum \mathrm{Cov}(X_i, X_j)$

k-th Moment (Raw):
$\mu_k = \mathrm{E}[X^k] = \int_{-\infty}^{\infty} x^k f(x) dx$

k-th Central Moment: $\mu_k^c = \mathrm{E}[(X - \mathrm{E}[X])^k] = \int_{-\infty}^{\infty} (x-\mu)^k f(x) dx$

Standardised Moment is Central Moment normalised typically with division by an expression of the variance which renders the moment scale invariant.

The 1st to 4th moments of the standard normal distribution $N(0,1)$ are $0,1,0,3$.

Expectation/Mean:
$\mu = \mu_1 = \mathrm{E}[X]$

Variance:
$\mu_2^c = \mathrm{Var}[X]$

Skewness:
$\mathrm{E}\left[\left(\frac{X-\mathrm{E}[X]}{\mathrm{Sd}[X]}\right)^3\right] = \frac{\mu_3^c}{(\mu_2^c)^{\frac{3}{2}}}$

Kurtosis:
$\mathrm{E}\left[\left(\frac{X-\mathrm{E}[X]}{\mathrm{Sd}[X]}\right)^4\right] = \frac{\mu_4^c}{(\mu_2^c)^2}$

Survival Function: $S(x) = 1 - F(x)$

Hazard Function: $h(x) = \frac{f(x)}{S(x)}$ roughly the conditional probability that the individual will die at time x given that it has survived until x.

CDF-Method: $Y = g(X)$ want $F_Y(y) = P[g(X) \leq y] = P[X \leq g^{-1}(y)] = F(g^{-1}(y))$

$f_Y(y) = f_X(g^{-1}(y))|(g^{-1})'(y)|$

Log-Likelihood Function: isomorphs a product of exponentials to a sum which one can differentiate to produce an extremum e.g.

Combinatorics And Discrete Casework

Dynamic Programming

Kelly Criterion: $\max\left(\sum P_i \ln(a_i)\right)$

Kelly Bet Ratio On A p Biased Coin At $1:1$ Odds: $2p-1$

Portfolio Optimisation Markowitz: on curve of expected returns and variance. Gradient, Lagrange Multipliers, set derivatives to 0.

Jensen: convex f,$a_i \geq 0$, $\sum a_i = 1$, $\sum a_i f(x_i) \geq f\left(\sum a_i x_i\right)$

Type I Error: Falsely Rejecting True Null Hypothesis

Type II Error: Failing To Reject False Null Hypothesis
Probability β
Power $1-\beta$

Data Science And Machine Learning

See Deep Learning Notes in my Github, Lazar repository, Notes, Computer Science, Deep Learning.

Linear Regression:
$X\beta = \hat{Y}$

X: the matrix of input vectors where each row is an observation, and each column is an input variable vector. One can append a 1 to the front of each such vector and thus produce the constant term β_0 in this simply expressed form. Understand instruments' measurement errors, precision, impacts, sensitivity, dynamic range, detection threshold, upper bound threshold, distortion, total distortion, total harmonic distortion, etc.
β: a column vector of the regression coefficients which are to be solved for in terms of gradients and derivatives.
Y: a column vector [can be matrix if multiple output vectors] of the response variable.
$\hat{Y}$: a column vector of the model's output for these input X values. Estimator, best fit under a metric on the training set, and sometimes in the literature refers to output predictions on a test set.

Ordinary Least Squares
Minimise Sum Of Squared Residuals:
$\text{SSR} = \text{RSS} = \sum(y_i - \hat{y}_i)^2$

$\hat{\beta}_1 = \frac{\sum(x_i-\bar{x})(y_i-\bar{y})}{\sum(x_i-\bar{x})^2}$

$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1\bar{x}$

Total Sum Of Squares:
$\text{SST} = \text{TSS} = \sum(y_i - \bar{y})^2$
$R^2 = 1 - \frac{\text{SSR}}{\text{SST}}$

Coefficients, Estimate, Standard Error, t Value= $\frac{\text{Estimate}}{\text{Standard Error}}$ [|t Value| $>$ 2], p Value [Tiny], Residual Standard Error On a Degrees Of Freedom, Multiple R^2, Adjusted R^2, F-Statistic On a And b Degrees Of Freedom, p Value.

`What are the 5 assumptions behind linear regression?`

Linear Relationship: The relationship between the independent and dependent variables should be linear. This can be tested using scatter plots which can reveal linear, logarithmic, square root, polynomial, non correlations, and other relations amongst variables.

Multivariate Normal: All the variables together should be multivariate normal. For all the variables to be multivariate normal each variable separately has to be univariate normal means a bell shaped curve. And any subset of variables should also be multivariate normal. This can be tested by plotting a histogram.

No Multicollinearity: There is little or no multicollinearity in the data. Multicollinearity happens when the independent variables are highly correlated with each other. Multicollinearity can be tested with a correlation matrix. Some people would think an absolute correlation coefficient of > 0.7 amongst 2 or more predictors indicates the presence of multicollinearity.

No Autocorrelation: There is little or no autocorrelation in the data. Autocorrelation means a single column data values are related to each other. In other words $f(x+1)$ is dependent on value of $f(x)$. Autocorrelation can be tested with scatter plots.

Homoscedasticity: This means "same variance". In other words, residuals are roughly equally symetrically normally distributed down the x-axis vertically off of the regression line. Homoscedasticity can also be tested using scatter plots.

There exist Python libraries gvlma, sklearn.

What does one do when each of these assumptions is violated?

Transformations can help when the homoscedasticity assumption, the linearity assumption, or normality is violated. The output/Y vector can be transformed too, not just input vectors. One might think autocorrelation in the time series setting is sometimes addressed via instead considering the Δ/discrete differences vector[s] as input instead of the raw source original.

How does one derive the closed form Ordinary Least Squares solution?

As the loss is convex the optimum solution lies at gradient 0. $(X^TX)^{-1}X^TY$. The goal is to minimise the cost function $J(\beta) = (y - X\beta)^T(y - X\beta)$. Expand and differentiate with respect to β.

What about weighted?

Gauss-Markov Theorem: the Ordinary Least Squares estimator has the lowest sampling variance within the class of linear unbiased estimators.

$S = \sum W_{ii} r_i^2, W_{ii} = \frac{1}{\sigma_i^2}$
Gradient Equations:
$-2\sum W_{ii}\frac{\delta f(x_i,\beta)}{\delta\beta_j} r_i = 0$
$\sum\sum X_{ij}W_{ii}X_{ik}\hat{\beta}_k = \sum X_{ij}W_{ii}y_i$
$(X^TWX)\hat{\beta} = X^TWy$

Non-Linear Least Squares Systems:
$(J^TWJ)\Delta\beta = J^TW\Delta y$

$\hat{\beta} = (X^TWX)^{-1}X^TWy$
Estimated Variance-Covariance Matrix
Error Propagation

How does one derive the closed form Ordinary Least Squares/L2 Ridge solution?

$(Y - X\beta)(Y - X\beta) + \lambda\beta^T\beta$
$X^TY = (X^TX + \lambda I)\beta$
$\beta = (X^TX + \lambda I)^{-1}X^TY$

What is the purpose of L2 Ridge, L1 Lasso? When does one use L2 Ridge versus L1 Lasso?

Reduce model complexity and prevent over fitting which may result from linear regression. Norm input vectors to mean 0 variance 1. In L2 Ridge regression, the cost function is altered by adding a penalty term of λ times square of the magnitude of the coefficients. The lower the constraint λ on the features, the more the model resembles linear regression. As L1 Lasso regularisation leads to coefficients of 0, not only does L1 Lasso regression help in reducing over fitting but it can help us with feature selection/engineering.

If one runs Ordinary Least Squares, L2 Ridge, L1 Lasso, what will the weights look like in each?

In L2 Ridge the weights will have slightly more normed lower values. In L1 Lasso even moreso with 0s as aforementioned.

How does one choose λ in L2 Ridge/L1 Lasso?

Cross validation. Consider k fold. Common values are 10, 5, and 3 for k but there exist maths to decide upon a k value. The key idea is to randomly split the dataset into k subsets/folds. And then for each fold execute a model upon the remaining data as training set and that fold as the test set. Aggregate these evaluation score/performance metrics in some way for the overall evaluation score/performance metric of the model. k fold cross validation, k fold cross validation with shuffle, stratified k fold cross validation, leave one out cross validation, repeated k fold cross validation, shuffle split cross validation, group k fold cross validation.

What is the relation between the following optimisation problems?

$\min\{||Y - X\beta||_2^2 + \lambda||\beta||_2^2 : ||\beta||_2 \leq \alpha\}$

$\min\{||Y - X\beta||_2^2 + \lambda||\beta||_1^2 : ||\beta||_1 \leq \alpha\}$

L2 Ridge and L1 Lasso bounded to ball and cube duals. Since the objective function is convex the L1 Lasso case will be on a vertex or boundary edge which is why 0 coefficients are produced rather than the nonzero of L2 Ridge. See Elements Of Statistical Learning.

Why is L2 Ridge regularisation equivalent to Gaussian prior?

Regularise the parameter β by imposing the Gaussian prior $N(\beta|0, \lambda^{-1})$. Hence, combining the likelihood and the prior we have $\prod N(y_n|\beta x_n, \sigma^2)N(\beta|0, \lambda^{-1})$. Taking the logarithm and dropping some constants one obtains $\sum -\frac{1}{\sigma^2}(y_n - \beta x_n)^2 - \lambda\beta^2 + c$ which is maximised with respect to β at the Maximum A Posteriori estimate for β. In this case but not the L1 Lasso case it is also a mean of the posterior for a suitable prior. See Elements Of Statistical Learning. And this is why the Gaussian prior is equivalent with L2 Ridge regularisation.

Why is L1 Lasso regularisation

equivalent to double exponential [Laplace] prior?

Analogous to previous.

When should one prefer gradient descent/stochastic gradient descent to solve linear regression rather than the closed form solution?

More computationally efficient. Consider matrix inversion and parameters sizes. If small, one can execute the closed form.

Gradient descent is an algorithm which functions in terms of a computed gradient and step size in the direction of a desired extremum of an objective function. And the key idea of stochastic gradient descent is to more rapidly descend into the target extremum or region via substantially decreasing the compute for each step. By randomly testing some points in the neighbourhood to roughly approximate a gradient e.g. when the objective function satisfies certain desiderata and sinking more compute will not dramatically improve the direction of a step.

One does linear regression on a dataset of size n. Then one duplicates each row in that dataset, so now one's dataset has size $2n$, and one does linear regression again. What happens to the regression coefficients, R^2, standard errors of the regression coefficients, the t score, etc.?

$$\boxed{\times 1, \times 1, \times \frac{1}{\sqrt{2}}, \times \sqrt{2}}$$

What is a good algorithm to do linear regression in a streaming setting? One is periodically observing new data and needs to quickly output the exact regression coefficients at any moment.

Maindonald describes a sequential method based on Givens rotations. See Belsley, Kuh, Welsh. Literature.

The Ordinary Least Squares solution is $(X^TX)^{-1}X^TY$. How does one compute this in a distributed setting, where X is $n \times p$ and Y is $p \times 1$ and $n >> p$?

Parallelise stochastic gradient descent. See Xiangrui Meng et al. and Zinkevich et al.

Parallelise Stochastic Gradient Descent as follows: Split the data across multiple machines. At each step, each local machine estimates the gradient using a subset of the data. All gradient estimates are passed to a central machine, which aggregates them to perform a global parameter update. The downside of this approach is that it requires heavy network communication, which reduces efficiency.

Partition the data evenly across local machines. Each machine solves the problem exactly for its own subset of the data, using a batch solver. Final parameter estimates from the local machines are averaged to produce a global solution. The benefit of this approach is that it requires very little network communication, but the downside is that the parameter estimates can be suboptimal.

Allow each local machine to randomly draw data points. Run Stochastic Gradient Descent on each machine. Finally, average the parameters across machines to obtain a global solution. Like [2], this method requires little network communication. But, the parameter estimates are better because each machine is allowed to access a larger fraction of the data.

`Explain frequentist and Bayesian statistics in one’s words.`

There is a sense in which they are both fundamentally about their algorithms, computations, and processes. Frequentism can mean plugging in some values into a paired test function in Python, R, or WolframAlpha. It is about a model for the underlying data generation process. Optimised null hypothesis and alternate hypothesis models, their associated SSR values, F statistic values, statistics, and a resultant p value. Often under Maximum Likelihood Estimation. Bayesianism being updating a prior distribution [suspicious axiomatics] through a likelihood function into a posterior distribution. In any case, both are common in contemporary academic statistics and mainstream science papers. One ought to be able to speak about some such papers [for any domain one claims to be interested in].

Derivatives Theory And Stochastic Calculus

See Concepts And Practice Of Mathematical Finance Notes, Key Points, Solutions files in my Github, Lazar repository, Notes, Algorithmic Trading.

f: theoretical derivative/option value.

S: price of underlying instrument.

σ: volatility of underlying instrument.

t: time.

r: interest rate.

Delta: $\Delta = \frac{\delta f}{\delta S}$

Gamma: $\Gamma = \frac{\delta^2 f}{\delta S^2}$

Theta: $\Theta = \frac{\delta f}{\delta t}$

Vega: $v = \frac{\delta f}{\delta \sigma}$

Rho: $\rho = \frac{\delta f}{\delta r}$

Standard Brownian Motion/Wiener Process:
$X(0) = 0$
Continuous Everywhere, Differentiable Nowhere
$X(t) - X(s) \sim N(0, |t - s|)$
$X(t + s) - X(t)$ is independent of $X(t)$

X_t is a Martingale with respect to the filtration F_t
$|X|^2 - t$ is a Martingale with respect to the filtration F_t

$\text{E}[dX] = 0$
$\text{E}[dX^2] = dt$
$\lim_{dt \to 0} dX^2 = dt$
Discrete Approximation:
$dX = \phi\sqrt{dt}$
Where $\phi \sim N(0, 1)$
dX is $O(dt^{\frac{1}{2}})$
$dtdX$ is $O(dt^{\frac{3}{2}})$

Ito Product Rule:
If $dX_t = \alpha dt + \beta dW_t$ and $dY_t = \gamma dt + \lambda dW_t$:
$d(X_t Y_t) = X_t dY_t + Y_t dX_t + dXdY$
$= X_t dY_t + Y_t dX_t + \frac{1}{2}\beta\lambda dt$

Stochastic Differential Equations:
$dS = f(t, S)dt + g(t, S)dX_i$
$dS_i = f_i(t, S_0, \ldots, S_n)dt + g_i(t, S_0, \ldots, S_n)dX_i$
Where f is the drift, g is the diffusion.

Ito's Lemma And Basic Stochastic Integration:
For $F(X_t)$:
$dF = \frac{dF}{dX}dX_t + \frac{1}{2}\frac{d^2F}{dX^2}dt$
$F(X_t) = F(X_0) + \int_0^t \frac{dF}{dX}dX_\tau + \frac{1}{2}\int_0^t \frac{d^2F}{dX^2}d\tau$
For $F(X_t, t)$:
$dF = \frac{\delta F}{\delta X}dX_t + \left(\frac{\delta F}{\delta t} + \frac{1}{2}\frac{\delta^2 F}{\delta X^2}\right)dt$
$F(X_t, t) = F(X_0, 0) + \int_0^t \frac{\delta F}{\delta X}dX_\tau + \int_0^t \left(\frac{\delta F}{\delta t} + \frac{1}{2}\frac{\delta^2 F}{\delta X^2}\right)d\tau$

Forward Kolmogorov:
$\frac{\delta p}{\delta t'} = \frac{1}{2}\frac{\delta^2}{\delta y'^2}(B(y', t')^2 p) - \frac{\delta}{\delta y'}(A(y', t')p)$
Brownian Motion With Drift:
$dS = \mu dt + \sigma dX$

Vasicek:
$dS = \gamma(\bar{r} - r)dt + \sigma dX$
Solution:
$p(S, t; S', t') =$

$\frac{1}{\sigma S' \sqrt{2\pi(t'-t)}} e^{-\frac{(\log\left(\frac{S}{S'}\right)+(\mu-\frac{1}{2}\sigma^2)(t'-t))^2}{2\sigma^2(t'-t)}}$

Geometric Brownian Motion (Lognormal):
$dS = \mu S dt + \sigma S dX$
$\frac{dS}{S} = \mu dt + \sigma dX$

Cox, Ingersoll, Ross:
$dS = (v - \sigma S)dt + \sigma S^{\frac{1}{2}} dX$

Martingale:
$E[M_{t+1}|F_t] = M_t, \forall 0 \leq s \leq t$
Supermartingale:
$E[M_{t+1}|F_t] \leq M_t$
Submartingale:
$E[M_{t+1}|F_t] \geq M_t$

Radon-Nikodym Theorem:
$Q(A) = \int_A \left(\frac{dQ}{dP}\right) dP$ where $\frac{dQ}{dP}$ is the Radon-Nikodym derivative.

Ito Integrals Are Martingales:
$\mathrm{E}\left[\int_0^T g(t, X_t) dX_t\right] = 0$

Martingale Representation Theorem:
If M is a Martingale, there exists $g(t, X)$ such that
$M_T = M_0 + \int_0^T g(t, X) dX_t$

Fubini:
$\mathrm{E}\left[\int_0^T f(X_t) dt\right] = \int_0^T \mathrm{E}[f(X_t)] dt$

Exponential Martingale:
$M(t) = e^{S_t + f(t)}$ where
$f(t) = -(\mu + \frac{1}{2}\sigma^2)t$

Properties Of Ito Integrals:
Linearity:
$\int_0^T (\alpha f(t) + \beta g(t)) dX_t =$
$\int_0^T \alpha f(t) dX_t + \int_0^T \beta g(t) dX_t$ Isometry:
$\mathrm{E}\left[|\int_0^T f(t) dX_t|^2\right] = \mathrm{E}\left[\int_0^T |f(t)|^2 dt\right]$
Martingale:
$\mathrm{E}\left[\int_0^T f(t) dX_t | F_s\right] = \int_0^s f(t) dX_t$

Fundamental Asset Pricing Formula:
Value =
$\mathrm{E}^{\text{Measure}}$[PV(Expected Cash Flows)]

Risk-Free Asset:
$dB_t = rB_t dt, B(0) = B_0$
$B(t) = B_0 e^{rt}$

Underlying S:
$dS_t = \mu S_t dt + \sigma S_t dX, S(0) = S_0$
$S(t) = S_0 e^{\mu t - \frac{1}{2}\sigma^2 + \sigma X_t}$

Removing The TVM:
$S^*(T) = \frac{S(T)}{e^{rt}}$
$S^*(t) = S_0^* e^{(\mu - r - \frac{1}{2}\sigma^2)t + \sigma X_t}$
$dS^* = (\mu - r)S^* dt + \sigma S^* dX$

Self-Financing Portfolios:
Trading Strategy:
$\phi_t = (\phi_t^S, \phi_t^B)$ Processes
Self-Financing Portfolio: No In/Out Flows:
Value: $V_t(\phi) = \phi_t^S S_t + \phi_t^B B_t, \forall t \in [0, T]$
$V_t(\phi) = V_0(\phi) + \int_0^t \phi_u^S dS_u + \int_0^t \phi_u^B dB_u$
Arbitrage Opportunity:
$V_0(\phi) = 0$
With $P(V_T(\phi) > 0) > 0$ and
$P(V_T(\phi) < 0) = 0$.

Novikov Condition:
$\mathrm{E}\left[e^{\frac{1}{2}\int_0^T \theta_s^2 ds}\right] < \infty$
$M_t^\theta = e^{(-\int_0^t \theta_s dX_s - \frac{1}{2}\int_0^t \theta_s^2 ds)}$ is a Martingale.

Girsanov's Theorem:
$\frac{dQ}{dP} = e^{(-\int_0^t \theta_s dX_s - \frac{1}{2}\int_0^t \theta_s^2 ds)}$

$X_t^Q = X_t^P + \int_0^t \theta(s)ds$
Provides an expression for the Radon-Nikodym derivative.
Gives an explicit correspondence between P and Q in terms of their Brownian motion.
Assume θ and check that it satisfies the Novikov condition. Then we have the Radon-Nikodym derivative, and we can change measures.

Doleans/Stochastic Exponential:
$\epsilon\left(\int_0^t \theta_s dX_s\right) =$
$\exp\left(\int_0^t \theta_s dX_s - \frac{1}{2}\int_0^t \theta_s^2 ds\right)$
$X_t^Q = X_t^P - \int_0^t \theta(s)ds$

Feynman-Kac Equivalence:
PDE: $\frac{\delta V}{\delta t} + \mu\frac{\delta V}{\delta S} + \frac{1}{2}\sigma^2\frac{\delta^2 V}{\delta S^2} - rV = 0, V(T,S) = G(S)$
$dS_t = \mu(t,S_t)dt + \sigma(t,S_t)dX_t$
$\iff$
Expectation:
$V(t,S_t) = e^{-r(T-t)}\mathrm{E}[G(S_T)|F_t]$

Examples

I am merely mortal. But I have done thousands of questions from GlassDoor and attempted to include sharp tasks, thoughts, cases, notes, and generalisations. Some of the later writeups are handwaved and not representative of good interview oration.

`Why dost thou wisheth to be an algorithmic quantitative trader?`

It will be fun and interesting. I have read a number of books on related topics, and I know people like me who have enjoyed careers in trading.

`Why our firm?`

[X, Y, Z specifics about this particular firm. Can be details from the public written record such as assets under management, number of employees, reputation, culture. Demonstrate that one somewhat comprehends what they do.]

[If one is an undergraduate studying stochastic calculus and mathematical finance, a firm may prefer the candidate with the physics PhD, despite one's current edge in some technical skills. Perhaps the firm's hiring process values "intelligence" in the real concrete sense of growth potential. Perhaps that candidate has other skills, is older, and is in a different financial life position. In any case, note that very intelligent humans often walk the path of a pure Science, Technology, Engineering, and Maths PhD at an elite R1. Some kids who do maths contests might assume that 0.95 of people in trading know what "Putnam Honourable Mention" means in terms of rank. That value may be closer to 0.15. In finance, one may observe more intelligent people, and also comparably intelligent people, who make insights, produce papers, and generate income. Also imagine how one will feel 24 months into the job. You know much more about the actual tasks and technical skills as well as the correlates. How much do you care now about puzzles? What rank is adequately meritorious to warrant an interview? How much does rank really matter? The fun Computer Science textbooks really do matter quite a lot and contain much directly useful content and information!]

`Weaknesses?`

Frankly, I am not quite as accurate as I would like to be. I know people who are less error prone. Sometimes I make many edits and improvements rather than acing something on my first try. I solve many tasks, and do not see a technique or habit to build. I used to be worse at communicating in technical writing, but thought and practice have helped me get better.

Off the dome, list 5 things you are not.

A robot, monkey, dancer, funky, as sharp as I aspire to being tomorrow and in the future.

[To run these programs, one can copy and paste the code, and then call the "replace all" function in a text editor from 4 space characters to 1 tab character prior to execution.]

A taxi in New York City costs 20 and a tuxedo rental 100.

There may be a relatively low cost to entry and quite a few people in New York City such that the equilibrium in the former market... one can discuss cars, gasoline costs, culture, the convenience latency factor on users, etc. as for tuxedos the number of firms engaged in that business is relevant as is the matter of tuxedos wearing out over time, costing potentially quite a lot up front to buy, storefront street level property rent if not delivery in New York City, perhaps a riskier or more volatile demand side, and even the notion of say "going out of fashion". I opted to bring up the sort of counterfactual option to the consumer being say a brand new Dziordzio Armani 2022 tuxedo.

Say you are holding a Vickrey auction where the maximal bidder pays the second maximal bidder's stated price point with n people who each cost 10 to recruit and will value the good at a value uniform in $[2000, 3000]$. What is the optimal number n to recruit in order to maximise the expected earnings?

A Game Theoretically Optimal strategy on the people is to simply bid their valuation. Maximise $3000 - 1000 \cdot \frac{2}{n+1} - 10n$ at $n = \boxed{13}$. This due to the expectation of the order statistic e.g. m of n uniform random variables in $[0, 1]$ being simply given by $\frac{m}{n+1}$ in the division between 2 simple Probability Density Function integrals.

Find the first instance of a given target value x in a sorted array.

Binary search in $O(\log(n))$. Use a while loop with left, right, and middle index variables. The comparison needed to handle an input case such as $[1, 2, 2, 2, 2, 3]$ checks if the value at the middle index is less than the target value. If so, this index is too small, to the left of the target index. Or, if the value to the left of the middle index is at least the target value, in which case this index is too large, to the right of the target index. Otherwise, this is the desired index to output.

Python Implementation:

```
values = [1, 2, 2, 2, 2, 3]
targetvalue = 2
if values[0] == targetvalue: # To simplify logical evaluation:
    targetindex = 0
else:
    left = 1
    right = len(values) - 1
    middle = (left + right) // 2
    while left < right:
        if values[middle] < targetvalue:
            left = middle + 1
        elif values[middle - 1] >= targetvalue: # Right here.
            right = middle - 1
        else:
            break
        middle = (left + right) // 2
    targetindex = middle
print(targetindex)
```

Given an unsorted array containing integers $1, 2, 3, \ldots, n$ with one number missing, find it.

Sum the arithmetic series using a long long to avoid integer overflow and produce $\frac{n(n+1)}{2}$ and then iterate through the array subtracting off until one is left with the remainder which is the missing number. If desired, one can do this all modulo n instead, being sure to associate 0 in $\mathbb{Z}/n\mathbb{Z}$ with the underlying value n.

Python Implementation:

```
values = [1, 6, 4, 7, 2, 5]
missingnumber = (len(values) + 1) * (len(values) + 2) // 2
for value in values:
    missingnumber -= value
print(missingnumber)
```

Find all triples of elements $[a, b, c]$ in an array such that $a + b = c$.

One can call the default library array sorting function in $O(n \cdot \log(n))$ and then for each element c execute a 2 pointers search in $O(n)$, thus producing an $O(n^2)$ algorithm. Initialise the pointers at the ends of the sorted array and while the left pointer is to the left of the right pointer, if the current sum is less than the target sum c, iterate the left pointer $++$, if the current sum is more than the target sum c, iterate the right pointer $--$, if the current sum is equal to the target sum c then add/output this triplet and $++$ the left pointer, $--$ the right pointer, and continue. This is asymptotically optimal. $O(n^2)$ is necessary. As a lower bound consider the arithmetic progression $[1, 2, 3, ..., n]$ which induces
$0 + 0 + 1 + 1 + 2 + 2 + \cdots = \left\lfloor \frac{n^2}{4} \right\rfloor$ such triplets.

Python Implementation:

```
# Handles case on distinct values.
values = [1, 8, 4, 6, 3, 5, 2, 7]
values.sort()
triplets = []
for value in values:
    left = 0
    right = len(values) - 1
    while left < right:
        current = values[left] + values[right]
        if current < value:
            left += 1
        elif current > value:
            right -= 1
        else:
            triplets.append([values[left], values[right], value])
            left += 1
            right -= 1
print(triplets)
```

One is guarding 100 rational murderers in a field, and one has a gun with 1 bullet. If any of the murderers has a nonzero probability of surviving, he will attempt to escape. If a murderer is certain of death, he will not attempt an escape. How does one stop them from escaping?

Say one will shoot the murderer with the lowest number who attempts to escape.

It is common knowledge that murderer number 1 will now not attempt escape, and thus neither will numbers $2, 3, 4, \ldots, 100$ inductively.

`I have a bag with 1000 coins in it. One of them is a double headed coin, the other 999 are fair coins. I pick 1 coin from the bag at random, and flip it 10 times. It comes up heads all 10 times. What is the probability that I have selected the double headed coin?`

The prior odds ratio is $1 : 999$ and we update through a $1 : \frac{1}{1024}$ likelihood function odds ratio to obtain $1024 : 999$ posterior odds ratio is a $P = \boxed{\frac{1024}{2023}}$.

[If asked the variant where one observes a coin from some dude's pocket flip 100 Heads in a row, do not say "prior", just say one's credence upon this observation is that the coin is double headed. In what context precisely is this observation taking place? Is it a metaphor for an asset price movement? In terms of actual numerics there is not really a great setting in which to do this. If a firm showed me a sequence of bits 1 by 1 and I made ≈ 100 observations sort of in line with a true source of pseudorandom bits and then it hit 12 in a row I'd probably feel suspicious that a phase transition in their process had taken place and by 16 I would be atmospherically suspicious. "Something's wrong, I can feel it". If pressed, say one is bounded and always has credence mass on the unknown knowns ["ideology"/"water"]. Clarify that even if one had never before thought of a double headed coin, one's surprise, the information theoretic surprise function, would be so high that the notion of a double headed coin would come into one's mind. Upon such an observation it would propagate into known known territory that the coin was double headed. Another out is to discuss actual numerics of double headed coins in one's life. Make a The Dark Knight by Christopher Nolan character Two-Face "I don't rely on chance. I make my own luck" reference. Just kidding, never make references at work. Stick to the object level maths structures.]

`We play a game: I pick a number` n `from 1 to 100. If one guesses my number correctly, one wins` n `and otherwise one wins 0. How much would one pay to play this game?`

In the Nash Equilibrium, the mixed strategy is set such that I am indifferent between each number and thus each number is set with $P(n) = \frac{\frac{1}{n}}{H_{100}}$ so the fair price linear expected value of this game, from me picking 1 e.g., is $\boxed{\frac{1}{H_{100}}}$ where

H_{100} is the 100th Harmonic Number $1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{100}$.

`What is the probability the first workday of a month is a Monday?`

The first day of the month is roughly uniformly distributed and this event happens if and only if the first day is a Saturday, Sunday, or Monday with probability roughly $\boxed{\frac{3}{7}}$.

[Red flag. All input details matter. “Roughly”, because of contemporary Gregorian calendar details. Can start out a response with something like “I observe that without ‘work’ the answer is one seventh so that bit of information ‘work’ must matter, upon which I more closely examine....”.]

`A submarine starts at some integer point; it moves a constant number of integers each turn. Once per turn one can lob a missile at some point on the integer line. Can one give an algorithm that will hit the submarine in a finite number of turns?`

$\boxed{\text{Yes}}$. The [initial starting point,speed] cases could be plotted in a coordinate lattice and to be sure we could spiral outwards from the origin, ensuring to hit each submarine case 1 by 1. On the first shot at time step $t = 0$ we target 0 for the case $[0, 0]$, on the second shot at time step $t = 1$ we target 1 for the case $[1, 0]$, on the third shot at time step $t = 2$ we target -1 for the case $[1, -1]$, etc. Thus we hit the submarine in the max magnitude coordinateth layer which is $O(n^2)$ in the max magnitude coordinate.

`Estimate the number of pennies in a stack the height of the Sears Tower.`

A roll of pennies is roughly 50 pennies to 7.5cm. One way to measure the height may involve similar triangles, measuring the height and shadow of a flag pole and the shadow of the tower. Or skip the flag pole and use live satellite imagery. Deduce the angle and tangent ratio geographically and astronomically from a chronological data point. Eyeballing, it seems around 400m tall and I know that the Burj Khalifa is around 800m tall so historically this checks out. This gives $\boxed{\approx 270000}$ pennies.

[Appear learned. This is a fantastic opportunity to incorporate cool Wikipedia readings, contemplations, and creative ideas related to measurements. As well as some precomputed reference points.]

`Outside a room there are 4 switches, and in the room there is a light bulb one can not see. 1 of the switches controls the light. The task is to find out which switch. One may turn any number of switches on or off, any number of times one wants. But one may only enter the room once.`

Turn on switches A and B for a while. Then turn B off and C on. Now enter the room and touch the bulb. We have 2 bits, an On/Off and Warm/Cool bit, with which to deduce the true switch. If one has quite a lot of time then one can do a 5th by turning on switch E for say 1 year and deduce it if the light bulb is such that one can see if it has burnt out.

`Give me an off the dome estimate of [the probability]...`

[I suggest 2 digits of precision. It sounds better.]

`Let us bet on [something, perhaps vague]...`

Please, write down the terms of resolution for this bet, as that is the precise object upon which we are betting. And the proposed escrow. I use the AdBlock extension on the Google Chrome browser to hide the Cascading Style Sheets elements associated with titles on betting sites, Metaculus, and other predictions markets. This is because they can cause one to exhibit the anchoring effect cognitive bias on something dumb.

`Complete this integer matrix so that its inverse is integer. [The GlassDoor post did not contain particulars about the task matrix.]`

An integer matrix is unimodular if and only if it has determinant ± 1. The general linear group, Cramer's rule, and note that if a row or column is complete and has GCD $\neq 1$ this can not be done. Criteria, desiderata, algorithms... if one can execute row/column swaps to produce a triangular matrix with diagonal entries of ± 1 then it is trivial. A way to try and achieve this is to sort the rows and columns based upon the number of elements which appear in them. Swaps multiplying the determinant by ± 1, and thus not changing the condition. This produces something but not necessarily that thing. If all but 2 entries in a row are filled then a certain GCD $= 1$ with expansion by minors implies that there exist 2 integers which will produce a linear combination with the rest and form 1 as desired. Inspection on expansion by minors works in some cases. If we have 3 rows which are identically filled in except for 1 missing column then we deduce these 3 rows are linearly dependent no matter what our choices, and thus the matrix has determinant 0, so these cases are impossible.

[This is an example of what I might say if I do not on sight or instantly crack such a task in an interview setting.]

Given the price of n stocks and the future price, find the maximum return that can be achieved under a certain budget. If one has 4 dollars, 4 stocks are currently worth $[1, 1, 1, 4]$, and in the future, these 4 stocks will appreciate to $[2, 2, 2, 6]$. At this time, if one buys the first 3 stocks, spend 3 dollars to reach the maximum profit of 3 dollars $3 = (2 - 1) + (2 - 1) + (2 - 1)$. n is limited to 1 to 300, and the budget is limited to 1 to 30000.

This is an Nondeterministic Polynomial Time Complete task in constrained integer linear programming. So one could find a performant approximation algorithm implementation. However, for these input bounds, an $O(nm)$ dynamic program works.

Python Implementation:

```python
stocks = [1, 1, 1, 4]
appreciated = [2, 2, 2, 6]
profit = [0 for a in range(len(stocks))]
for a in range(0, len(stocks)):
    profit[a] = appreciated[a] - stocks[a]
budget = 4
maxthusfar = [0] * (1 + budget)
for a in range(0, len(stocks)):
    for b in range(budget, stocks[a] - 1, -1):
        if maxthusfar[b] < maxthusfar[b - stocks[a]] + profit[a]:
            maxthusfar[b] = maxthusfar[b - stocks[a]] + profit[a]
print(max(maxthusfar))
```

What is i^i?

$i^i = \left(\cos\left(\frac{\pi}{2}\right) + i\sin\left(\frac{\pi}{2}\right)\right)^i = \left(e^{i\cdot\frac{\pi}{2}}\right)^i = \boxed{e^{-\frac{\pi}{2}}}$.

What is a good strategy to overhead press "95"kg without actually being able to lift 95kg?

Pick 2 of the “40”kg plates and a “women’s” “15”kg barbell. This maximises the probability of the weight being under one's threshold. The variance in the ratio of the true underlying to written weight may be larger. The thinner bar and potentially superior knurling will enhance grip and position relative to body, as well as stability due to moment around gravicenter of the "95"kg mass which could truly be 85kg.

[Unironically though, when gambling against the house's edge it often makes sense to dynamically program in conjunction with never betting more than needed and hoping to maximise the probability of winning by minimising the number of bets e.g. by maximising the bet sizes.]

`What is the expected number of die rolls until the first instance of 6 sixes in a row?`

This is a Markov process with the relevant states being on a streak of 0 sixes, 1 six, 2 sixes, etc. in a row. Each state either leads to the next state in the chain or recurses to the starting null state except for the termination/hitting. Casework on number of sixes prior to to this next recurrence gives:

$X = (1-p)(X+1) + p(1-p)(X+2) + \cdots + p^{n-1}(1-p)(X+n) + p^n(n)$
$= \frac{1+p+p^2+...p^{n-1}}{p^n} = \frac{1-p^n}{p^n(1-p)} = \boxed{55986}$

For a more general similarly structured Markov chain one could iterate down the chain, keeping a partial probability stored, and produce the linear equation in terms of a constant and coefficient to resolve in $O(n)$.

How many races will one play in Mario Kart Double Dash prior to getting an all cups maximum score of 160 for acing through the 16 races, supposing that one quits and restarts any time one loses a race? Global minimum of 16 for me, I never lose. How many interview rounds will one go through prior to signing an offer? Yikes.

Python Implementation:

```
# Include a positivity check here to ensure finitude.
transitionPs = [1 / 6, 1 / 6, 1 / 6, 1 / 6, 1 / 6, 1 / 6]
Xcoefficient = 0
constantterm = 0
Pofreachingstate = 1
for a in range(1, 1 + len(transitionPs)): # Not optimal performance.
    Xcoefficient += Pofreachingstate * (1 - transitionPs[a - 1])
    constantterm += a * (Pofreachingstate * (1 - transitionPs[a - 1]))
    Pofreachingstate *= transitionPs[a - 1]
constantterm += (len(transitionPs)) * Pofreachingstate
print((constantterm) / (1 - Xcoefficient))
```

In terms of a system of linear equations:

$$
\begin{aligned}
&55986 = A = 1 + \tfrac{5}{6} \cdot A + \tfrac{1}{6} \cdot B \\
&55980 = B = 1 + \tfrac{5}{6} \cdot A + \tfrac{1}{6} \cdot C \\
&55944 = C = 1 + \tfrac{5}{6} \cdot A + \tfrac{1}{6} \cdot D \\
&55728 = D = 1 + \tfrac{5}{6} \cdot A + \tfrac{1}{6} \cdot E \\
&54432 = E = 1 + \tfrac{5}{6} \cdot A + \tfrac{1}{6} \cdot F \\
&46656 = F = 1 + \tfrac{5}{6} \cdot A
\end{aligned}
$$

In terms of the usual matrix inversion:

$$
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & \frac{5}{6} & \frac{1}{6} & 0 & 0 & 0 & 0 \\
0 & \frac{5}{6} & 0 & \frac{1}{6} & 0 & 0 & 0 \\
0 & \frac{5}{6} & 0 & 0 & \frac{1}{6} & 0 & 0 \\
0 & \frac{5}{6} & 0 & 0 & 0 & \frac{1}{6} & 0 \\
0 & \frac{5}{6} & 0 & 0 & 0 & 0 & \frac{1}{6} \\
\frac{1}{6} & \frac{5}{6} & 0 & 0 & 0 & 0 & 0
\end{bmatrix}
$$

$(I - Q)^{-1}$ can be computed with the following terse code.

WolframAlpha Implementation:

```
Inverse {{1/6,-1/6,0,0,0,0},{-5/6,1,-1/6,0,0,0},{-5/6,0,1,-1/6,0,0},
{-5/6,0,0,1,-1/6,0},{-5/6,0,0,0,1,-1/6},{-5/6,0,0,0,0,1}} =
```

$$
\begin{bmatrix}
46656 & 7776 & 1296 & 216 & 36 & 6 \\
46650 & 7776 & 1296 & 216 & 36 & 6 \\
46620 & 7770 & 1296 & 216 & 36 & 6 \\
46440 & 7740 & 1290 & 216 & 36 & 6 \\
45360 & 7560 & 1260 & 210 & 36 & 6 \\
38880 & 6480 & 1080 & 180 & 30 & 6
\end{bmatrix}
$$

This produces the expected number of hittings and row sums yield the aforementioned values.

Find a vector that forms the same angle with n vectors in $\mathbb{R}^n$.

One can norm the input to unit vectors on the unit sphere. Same angle implies same cosine implies same dot product so that $w \cdot (v_j - v_1) = 0$, and w is orthogonal to $\text{span}(v_j - v_1)$ which can be done in $O(n)$ with Gram-Schmidt.

Given 2 datasets X and Y, we run 2 linear regressions to obtain $y \sim ax + b$ and $x \sim cy + d$. What are the bounds on ac?

Without loss of generality, both datasets have mean 0. Then, by the covariance definition of slope with the Cauchy-Schwarz inequality on the sums we obtain bounds of $\boxed{0 \leq ac \leq 1}$.

Give an example of 2 variables that are uncorrelated but dependent.

$X \sim N(0,1)$ and $Y = X^2$ works as $\text{Cov}(X, Y) = \text{E}[X^3] - \text{E}[X] \cdot \text{E}[X^2] = \boxed{0}$.

Choose $(n-1)$ points uniformly randomly on a line segment and break the segment at those points. What is the probability that the resulting n segments form an n-gon?

Isomorphic with n uniform random points on a unit circle not all lying on a semicircle thus by independence $\boxed{1 - \frac{n}{2^{n-1}}}$. To clarify, the probability that all of the other points lie on the semicircle clockwise from the say 3rd selected point is $\frac{1}{2^{n-1}}$ and these n such events are mutually exclusive with probability 1 there is at most 1 unique counterclockwise most point inducing such a semicircle upon which all the points lie.

Suppose 3 assets A, B, and C are such that $\text{Corr}(A, B) = 0.9$ and $\text{Corr}(B, C) = 0.8$. Is it possible for $\text{Corr}(A, C) = 0.1$?

$\boxed{\text{No}}$, the correlation matrix must be positive semidefinite but the determinant is negative.

WolframAlpha Implementation:

```
Determinant {{1,0.9,0.1},{0.9,1,0.8},{0.1,0.8,1}}
```

$$\begin{vmatrix} 1 & 0.9 & 0.1 \\ 0.9 & 1 & 0.8 \\ 0.1 & 0.8 & 1 \end{vmatrix} = -\tfrac{79}{250} < 0$$

Suppose a collection of n random variables have all pairwise correlations equal to ρ. Find, with proof, the range of possible values of ρ.

A matrix is positive semidefinite if and only if all eigenvalues are non negative.

$$\begin{bmatrix} 1 & \rho & \rho & \rho & \rho \\ \rho & 1 & \rho & \rho & \rho \\ \rho & \rho & 1 & \rho & \rho \\ \rho & \rho & \rho & 1 & \rho \\ \rho & \rho & \rho & \rho & 1 \end{bmatrix} \approx$$

$$\begin{bmatrix} 1 & \rho & \rho & \rho & \rho \\ \rho-1 & 1-\rho & 0 & 0 & 0 \\ \rho-1 & 0 & 1-\rho & 0 & 0 \\ \rho-1 & 0 & 0 & 1-\rho & 0 \\ \rho-1 & 0 & 0 & 0 & 1-\rho \end{bmatrix} \approx$$

$$\begin{bmatrix} 1+(5-1)\rho & 0 & 0 & 0 & 0 \\ \rho-1 & 1-\rho & 0 & 0 & 0 \\ \rho-1 & 0 & 1-\rho & 0 & 0 \\ \rho-1 & 0 & 0 & 1-\rho & 0 \\ \rho-1 & 0 & 0 & 0 & 1-\rho \end{bmatrix}$$

In this case $\lambda_1 = 1 + (n-1)\rho \geq 0$ with multiplicity 1 and $\lambda_2 = 1 - \rho \geq 0$ with multiplicity $n-1$ thus one obtains the inequality bounds $\boxed{-\dfrac{1}{n-1} \leq \rho \leq 1}$.

Give an example of a distribution with infinite variance.

$\text{Var}(X) = \text{E}[X^2] - \text{E}[X]^2$ is infinite for $X = \pm 1$ with $P = \frac{1}{4}$, $X = \pm 2$ with $P = \frac{1}{8}$, $X = \pm 4$ with $P = \frac{1}{16}$ etc. as $\text{E}[X] = 0$ and $\text{E}[X^2] = \frac{1}{4} + \frac{1}{2} + 1 + 2 + \ldots$. This construction can be modified to work for any $\epsilon > 0$ to produce a symmetric mean 0 distribution with infinite $\text{E}[|X|^\epsilon]$.

[The Cauchy distribution $\text{PDF}(x) = \frac{1}{\pi(1+x^2)}$ from Solutions for a common continuous example but similarly some symmetric example with sufficient extremal mass is a key idea.]

What is the cumulative distribution function and probability density function of the kth order statistic of n variables from an arbitrary probability function?

The cumulative distribution function $P[X_k \leq x]$ is the probability that the kth trial is to the left of x if and only if at least k of the n trials are to the left of x. This can be computed as a casework on precisely how many trials are to the left of x using binomials and terms of the form $(F(x))^a$ and $(1 - F(x))^b$ and then differentiate to obtain the probability density function:

$$\boxed{k\binom{n}{k} f(x)(F(x))^{k-1}(1 - F(x))^{n-k}}$$

`How does one generate 2 random variables from` $N(0,1)$ `with correlation` ρ `if one has an` $N(0,1)$ `random number generator?`

$$\boxed{a_1 = b_1, a_2 = b_1\rho + b_2\sqrt{1 - \rho^2}}$$

Indeed the means are 0, the variances add due to independence and thus the variance of a_2 is 1, and $\text{Cov}(a_1, a_2) = \text{Cov}(b_1, b_1\rho + b_2\sqrt{1 - \rho^2}) = \text{Cov}(b_1, b_1\rho) = \rho$. In general using Cholesky decomposition one can generate correlated random variables following an n-dimensional multivariate normal distribution by decomposing the covariance matrix into $R^T R$ and using $X = \mu + R^T Z$ where Z is a vector of random $N(0,1)$ values. Alternately one can use singular value decomposition and produce $X = \mu + UD^{\frac{1}{2}}Z$.

`If the probability of observing at least 1 car on a highway during any 20 minute time interval is` $\frac{609}{625}$`, then what is the probability of observing at least 1 car during any 5 minute interval? Assume uniformity.`

Poisson. $1 - (1 - p)^4 = \frac{609}{625}$ so $p = \boxed{\frac{3}{5}}$.

`One is waiting for a bus at a bus station. The buses arrive at the station according to a Poisson process with an average arrival time of 10 minutes [`$\lambda = 0.1$`/minute]. If the buses have been running for a long time and one arrives at the bus station at a random time, what is one's expected waiting time? On average, how many minutes ago did the last bus leave?`

Poisson, symmetry, both are $\boxed{10}$ minutes. One is more likely to arrive during the longer gaps over the whole time period. When the arrivals of a series of events each independently follow an exponential distribution, the number of arrivals in an interval such as $[0, t]$ is a Poisson process. The expected value and variance in number of arrivals are both λt. The expected time for a general distribution is $\boxed{\dfrac{\mathrm{E}[X^2]}{2\mathrm{E}[X]}}$.

`Given 2 memoryless light bulbs with expected lifetimes` x `and` y `what is the probability that the first burns out prior to the second?`

$\boxed{\dfrac{y}{x+y}}$. Indeed, the unique probability distribution with the memoryless property is the exponential/geometric. Without calculus, observe that if one had 1 of each bulb and replaced them as they burn out, over the long run the expected value of the fraction which were of the first kind would be as claimed.

`One just bought 1 share of stock A and wants to hedge it by shorting stock B. How many shares of B should one short to minimise the variance of the hedged position? Assume that the variance of stock A's return is` σ_A^2`; the variance of B's return is` σ_B^2`, their correlation coefficient is` ρ`.`

Suppose that we short h shares of B. The variance of the portfolio return is $\mathrm{Var}(r_A - hr_B) = \sigma_A^2 - 2\rho h\sigma_A\sigma_B + h^2\sigma_B^2$. Compute the zero of the first derivative of the variance with respect to h: $-2\rho\sigma_A\sigma_B + 2h\sigma_B^2 = 0$ at $h = \boxed{\rho \cdot \dfrac{\sigma_A}{\sigma_B}}$ confirmed minimum by inspecting the second derivative $2\sigma_B^2 > 0$.

`There is a 0.5 probability that bond A will default next year and a 0.3 probability that bond B will default. What is the range of probability that at least 1 bond defaults and what is the range of their correlation?`

Inequality bounds $\boxed{[0.5, 0.8]}$ on maximal/minimal overlap union, intersection.

Formulae compute the range of correlation $\boxed{\left[-\sqrt{\dfrac{3}{7}}, \sqrt{\dfrac{3}{7}}\right]}$.

`Suppose one has 2 covariance matrices` A `and` B`. Is` AB `also a covariance matrix? What if` $AB = BA$`?`

$\boxed{\text{No}}$, symmetry is not assured. $\boxed{\text{Yes}}$, commuting matrices have the same eigenbasis, hence A and B can be simultaneously diagonalised by some matrix U. Thus it follows $AB = UD_1U^{-1}UD_2U^{-1} = UD_1D_2U^{-1}$. Thus the eigenvalues of AB are each a product of an eigenvalue of A and an eigenvalue of B. Thus all the eigenvalues are non negative and AB is also positive semidefinite.

`Define and enumerate some properties of a Brownian motion.`

$W(0) = 0$

The increments $W(t_1) - W(0), W(t_2) - W(t_1), \ldots, W(t_n) - W(t_{n-1})$ are independent and normally distributed $N(0, t_{i+1} - t_i)$.

$\mathrm{E}[W(t)] = 0$
$\mathrm{E}[W(t)^2] = t$
$W(t) \sim N(0, t)$
Martingale property $\mathrm{E}[W(t+s)|W(t)] = W(t)$
$\mathrm{Cov}(W(s), W(t)) = s, \forall 0 < s < t$
Markov property.

$Y(t) = W(t)^2 - t$ is a Martingale.

$Z(t) = e^{\lambda W(t) - \lambda^2 \cdot \frac{t}{2}}$ is a Martingale.

`What is the correlation of a Brownian motion and its square?`

By symmetry $\mathrm{E}[X] = 0$, $\mathrm{E}[X^3] = 0$, $\mathrm{Cov}(X, X^2) = \mathrm{E}[X^3] - \mathrm{E}[X] \cdot \mathrm{E}[X^2] = \boxed{0}$.

`Let` X `be a Brownian motion. What is the probability that` $X_1 > 0$ `and` $X_2 < 0$`?`

One can integrate the relevant multivariate joint normal probability density function, however, by symmetry $P[X_1 > 0] = \frac{1}{2}$, $P[X_2 - X_1 < 0] = \frac{1}{2}$, $P[|X2 - X1| > |X1|] = \frac{1}{2}$ thus $\boxed{\frac{1}{8}}$.

`What is the expected stopping time for a Brownian motion to reach either` a `or` $-b$ `and probabilities? What if` X `has drift` m `i.e.` $dX(t) = mdt + dW(t)$`?`

$\mathrm{E}[\text{Stopping Time}] = \boxed{ab}$. $P[\text{Hitting a}] = \boxed{\frac{b}{a+b}}$.

X is no longer Martingale, however it is still Markov. Applying Feynman-Kac with boundary conditions $P(a) = 1$ and $P(-b) = 0$ one obtains a homogeneous linear differential equation with the 2 real roots $r = 0, -2m$ and the solution

$\boxed{\dfrac{e^{2bm}-1}{e^{2bm}-e^{-2am}}}$. Alternately, apply the exponential Martingale.

`What is the expected value and variance of` $Y = |X|$ `for` $X \sim N(0,1)$`?`

$\mathrm{E}[Y] = \boxed{\sqrt{\dfrac{2}{\pi}}}$ and $\mathrm{Var}(Y) = \boxed{1 - \dfrac{2}{\pi}}$.

`What could be some issues if the distribution of test data is significantly different from the distribution of training data?`

The model may perform quite poorly. It was trained on 1 region of input space and its validity on another is questionable. There may be a phase transition and error terms in polynomials approximations can blow up.

`What are some ways to make a model more robust to outliers?`

Tree based model. Non parametric test.

`What are some differences one would expect in a model that minimises squared error L2, versus a model that minimises absolute error L1? In which cases would each error metric be appropriate?`

Bias, overfitting, underfitting, accuracy/recall/confusion matrix performance notions and desiderata. L1 may be considered more robust as it will overfit less to outliers. However it is less stable to minor input perturbation jumps. L2 is not as robust, however it is stable and always has 1 solution.

`L1 Lasso and L2 Ridge regularisation?`

Optimisation tasks. Norm all input vectors to mean 0 variance 1 then compute the weights/coefficients which minimise:

$\lambda \sum |\beta_i| + \mathrm{SSR}$
$\lambda \sum |\beta_i|^2 + \mathrm{SSR}$

L1 Lasso is inefficient on non sparse cases but produces sparse outputs and thus ignoring 0s means useful as feature selection tool. L2 Ridge is more computationally efficient due to analytical solutions but produces non-sparse outputs and thus is not as effective for feature selection. See optimal subset selection algorithms in Elements Of Statistical Learning. See Formulae for derivation of L2 Ridge solution.

`What error metric would one use to evaluate how good a binary classifier is? What if the classes are imbalanced? What if there are more than 2 groups?`

Evokes Chi Squared metrics from Applied Statistics. As well as the imbalanced Kaggle task where the user randomly selected from the larger class to produce a more balanced training set.

Confusion Matrix, false positive rate, type I error, false negative rate, type II error, true negative rate, specificity, negative predictive value, false discovery rate, true positive rate, recall, sensitivity, positive predictive value, precision, accuracy, F beta score, F1 score, F2 score, Cohen kappa, Matthews correlation coefficient, Receiver Operating Characteristic Curve, Area Under The Receiver Operating Characteristic Curve score, precision-recall curve, precision-recall Area Under The Receiver Operating Characteristic Curve, average precision, log loss, Brier score, cumulative gain chart, lift curve, lift chart, Kolmogorov-Smirnov plot, Kolmogorov-Smirnov statistics.

`What are various ways to predict a binary response variable? Can one compare 2 of them and tell me when one would be more appropriate? What is the difference between these? [Support Vector Machines, Logistic Regression, Naive Bayes, Decision Tree, etc.]`

Support Vector Machines
Logistic Regression
Linear Regression
Naive Bayes
Decision Tree
Random Forest
Extreme Gradient Boosting

`What is regularisation and where might it be helpful? What is an example of using regularisation in a model?`

L1 Lasso, L2 Ridge, etc. metrics. Various ways to penalise coefficients which are algorithmically produced. There is a sense in which it is helpful because humans discovered that these techniques work effectively for target datasets.

`Why might it be preferable to include fewer predictors over many?`

Simpler model, parsimony, better test accuracy, computational reasons, in a trading system better performance overall, less fluctuation due to minor deviations in training data, etc. more human legible models.

Autocorrelation Analysis?

Autocorrelation analysis is an important step in the Exploratory Data Analysis [EDA] of time series. The autocorrelation analysis helps in detecting hidden patterns and seasonality and in checking for randomness. It is especially important when one intends to use an Auto-Regressive Integrated Moving Average ARIMA model for forecasting because the autocorrelation analysis helps to identify the AR and MA parameters for the ARIMA model.

Auto-Regressive [AR] Model, Moving Average [MA] Model, Stationarity, ACF and PACF assume stationarity of the underlying time series. Stationarity can be checked by performing an Augmented Dickey-Fuller [ADF] test:

p-value > 0.05: the data is non stationary.
p-value ≤ 0.05: the data is stationary.

Stationary Process: a stochastic process whose unconditional joint probability distribution does not change when shifted in time. Consequently, parameters such as mean and variance also do not change over time. Assumption underlying many statistical procedures used in time series analysis, non stationary data are often transformed to become stationary. A common cause of violation of stationarity is a trend in the mean, which can be due either to the presence of a unit root or a deterministic trend. In the former case of a unit root, stochastic shocks have permanent effects, and the process is not mean reverting. In the latter case of a deterministic trend, the process is called a trend stationary process, and stochastic shocks have only transitory effects, after which the variable tends toward a deterministically evolving [non constant] mean.

Autocorrelation Function [ACF]:
Correlation between time series with a lagged version of itself. The correlation between the observation at the current time spot and the observations at previous time spots. The autocorrelation function starts a lag 0, which is the correlation of the time series with itself and therefore results in a correlation of 1.

Partial Autocorrelation Function [PACF]:
Additional correlation explained by each successive lagged term. The correlation between observations at two time spots given that we consider both observations are correlated to observations at other time spots.

Problem Definition, Data Collection, Data Preprocessing, Chronological Order And Equidistant Timestamps, Handling Missing Values, Resampling, Stationarity, Feature Engineering, Time Features, Decomposition, Lag, Exploratory Data Analysis, Autocorrelation Analysis, Cross Validation, Models, Models for Univariate Time Series, Naive Approach, Moving Average, Exponential Smoothing, ARIMA, Models For Multivariate Time Series, Vector Autoregression [VAR]

`Given training data on tweets and their retweets, how would one predict the number of retweets of a given tweet after 7 days after only observing 2 days worth of data?`

This task has a time series component as well as a cluster analysis component perhaps the Twitter data scientists know a thing or two about separate sects of Twitter users and would have a multi level model of sorts based upon some implicit classification such as the basketball discussions tweets line up in one way whereas the general news retweet counts decay more rapidly.

`How could one collect and analyse social media data to predict the weather?`

Odd task to be sure. The literature would suggest that meteorologists have solid models and predictions based upon public and historical data on actual weather inputs. But we could try and see if text Machine Learning Natural Language Processing produces results especially target words related to weather or maybe even things like upcoming events or people maybe going to rush the HEB grocery stores prior to a horrible winter weather storm.

`Given a database of all previous alumni donations to one’s university, how would one predict which recent alumni are most likely to donate?`

We might care more about predicting the quantitative donation size as well as the precise time of donation. In any case consult literature, perhaps some time series notions based upon graduation date, and even trawling public datasets which include employment, marriage, address, income, etc.

`How would one approach the design of a heatmap in Uber to recommend drivers where to wait?`

Consult literature. A naive strategy involves obtaining data by testing locations, and trying to infer as time goes on better locations. One might learn that certain sides of roads like near intersections right turns off of a big road rather than left are better and faster for the end phase when a driver and passenger meetup and then right turn back on to the big road. Thus instruct the passenger to walk to there for pickup e.g. in the shared mode Uber does this though an aware passenger is always free to choose intersection location on their priors about the Uber algorithm and drivers' distribution to minimise their Estimated Time Of Arrival. That net flicks Netflix task was mildly interesting I do not know how they decide what to put available next. This task statement is also a little vague, Uber has a big optimisation problem task and drivers have their own incentives like staying near their homes towards the end of the day. And so if this is about general ambient waiting prior to a matching being initiated there are other considerations.

[Any potentially non trivial observation of structure is better than none.]

`How would one build a model to predict a March Madness bracket?`

A priori when tasked with X I would consult the literature on X as an early step in strategy, perhaps following a little ideation. In this particular case I know that Dr. Yan Zhang of San Jose State University and the Summer Program On Applied Rationality And Cognition won 3rd in a Kaggle contest titled March Machine Learning Mania 2016 so I might start by trawling the corpus there.

`One wants to run a regression to predict the probability of a flight delay, but there are flights with delays of up to 12 hours that are really messing up one's model. How can one address this?`

Regression suggests predicting the quantitative delay. One could turn this into a logistic regression for the binary outcome variable of whether or not it was written down that there was a delay. Then this task isomorphs perhaps into computing a threshold boundary for classification. Another idea may be to transform the outcome vector consider the log or a model with an exponent variable. Consider threshold the input delay vector by replacing all delays of a parameter >3 hours with 3 hours.

`Variations on ordinary linear regression can help address some problems that come up working with real data. L1 Lasso helps when one has too many predictors by producing weights of 0. L2 Ridge regression can help with reducing the variance of one's weights and` predictions by shrinking the weights. Least absolute deviations or robust linear regression can help when one has outliers. Logistic regression is used for binary outcomes, and Poisson regression can be used to model count data.

Some of this source text is acceptable writing.

Write a function to calculate all possible assignment vectors of $2n$ users, where n users are assigned to group 0 [control], and n users are assigned to group 1 [treatment].

A nested for loops approach.

C++ Implementation:

```
for(a = 0; a < 2 * n; a++){
    for(b = a + 1; b < 2 * n; b++){ // etc.
        // Put a 1 at indices a, b, etc. in a vector.
        // Add this vector to output vector of vectors.
    }
}
```

Another approach is to call the default permutation iterator to produce such multinomials [can be useful].

C++ Implementation:

```
av = {0, 0, 0, 0, 0, 1, 1, 1, 1, 1};
do{
    avv.add(av);
}while(nextpermutation(all(av)));
```

Another approach if n is small is to loop through integers in a for loop and use their bit set if the number of 1 bits i.e. the popsizecount is n in the first $2n$ bits. This can be done by taking the bitwise and operator with the string of the final $2n$ bits being 1 e.g. $2^{2n} - 1$.

C++ Implementation:

```
b = (1 << (2 * n)) - 1;
```

```
for(a = 0; a <= b; a++){
    if(subsetsize(a & b) == n){
        // Process a’s bit representation into a vector.
        // Add this vector to output vector of vectors.
        // Or add a to a vector used later in bit form.
    }
}
```

Given a list of tweets, determine the $k = 10$ most used hashtags.

Thanks [Name Of Interviewer] for the pun, hint on “hash”. Perhaps this task depends on the underlying database, system structure and this first phase could be parallelised in general. $\approx O(\min(n, m + k \cdot \log(m), m \cdot \log(k)))$ is asymptotically optimal under memory constraints due to reading input. One can heapify, build heap, produce a maximum heap from an unsorted array/vector in $O(m)$ by placing all elements into the heap incorrectly and then heapifying [to do: implement]. Hash map from hashtags to integers. Process and if count is 0 insert and map to 1, else ++ the multiplicity counter. At the end for output, this reduces into the sub task of producing the k largest elements from an iteratable with m elements. One way is to iterate through keeping the 10 most used thus far in a reservoir of a heap/priority queue/order statistic set. Thus in the worst case runtime is $O(m \cdot \log(k))$, where m is the number of distinct hashtags, but randomised hash function insertion or an $O(m)$ shuffle can mitigate against adversarial input and worst case runtimes. A shuffle up front might lead to better average runtime algorithms which are somewhat insightful statistically about ignoring hashtags which are likely to be irrelevant based on some processing of multiplicities. If we knew that hashtags were each used at most 10^8 times, which is plausible in this particular case, we could loop through in $O(m)$ placing into a multiplicity vector which could then be looped through from the maximum downwards afterwards to produce the output. I need to clock the execution times of the compiled C of optimal implementations on relevant machines.

Python Implementation:

```
import heapq
tweets = ["#A", "#A #B #C"]
hashtags = {}
k = 10
for tweet in tweets:
    words = tweet.split(" ")
    for word in words:
        # I think hashtags appear <=1 time per tweet.
        # Unsure if they can appear mid tweet.
        # If only at back can process/parse from back, halt.
        if word[0] == ’#’:
            hashtag = word[1:]
            if hashtag not in hashtags:
                hashtags[hashtag] = 1
            else:
                 hashtags[hashtag] += 1

# O(m) Supposing Multiplicities < 10^8

multiplicitywas = [[] for a in range(100000000)]
for hashtag in hashtags:
    multiplicitywas[hashtags[hashtag]].append(hashtag)
a = 1
b = 100000000 - 1
while a <= k and b >= 0:
if len(multiplicitywas[b]) > 0:
        for topktag in multiplicitywas[b]:
            print(topktag, " With Multiplicity ", b)
            a += 1
            if a > k:
                break
    b -= 1

# General O(m * log(k))

topk = [[-1, a] for a in range(k)]
topkhashtag = [" " for a in range(k)]
heapq.heapify(topk)
for hashtag in hashtags:
    if topk[0][0] < hashtags[hashtag]: # Breaking ties here.
        heapq.heappush(topk, [hashtags[hashtag], topk[0][1]])
        topkhashtag[topk[0][1]] = hashtag
        heapq.heappop(topk)
```

```
while topk[0][0] == -1: # If < k Distinct Hashtags.
    heapq.heappop(topk)
for hashtag in topk:
    print(topkhashtag[hashtag[1]], " With Multiplicity ", hashtag[0])
```

Write an approximation algorithm for the Nondeterministic Polynomial Time Complete task of budgetary allocation constrained integer linear programming.

Beyond searching the literature for a performant implementation one ought to consider the dataset and modifications and optimisations which improve performance given the specifics of that dataset like perhaps real world budgetary allocation task datasets are more organic under some metric which implies there exist superior algorithms.

Write an approximation algorithm for the Nondeterministic Polynomial Time Complete task of computing the minimum Hamiltonian Cycle on a given point set.

Beyond searching the literature for a performant implementation one ought to consider the dataset and modifications and optimisations which improve performance given the specifics of that dataset like perhaps real world cities are more distributed than uniform random points in a unit square in $\mathbb{R}^2$ under a certain metric and this implies there exist superior algorithms for such organic datasets. Consider Markov Chain Monte Carlo for trading firms.

Write an algorithm that will produce a random sample of k elements from a stream of unknown size.

Maintain a reservoir vector of k elements and at time step $t > k$ randomly replace 1 element from the reservoir with the new element with probability $P = \frac{1}{t}$. This algorithm falls from base case and inductive construction.

Write an algorithm to compute $7n$.

C++ Implementation:

```
return (n << 3) - n;
```

Write a function to compute $P(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n = a_0 + x(a_1 + x(a_2 + x(a_3 + \dots)))$ for `arbitrary input` x`. Now write one which will print out` $P(x)$ `for all integers` $x \in [a, b]$`.`

The second form enables less multiplication operations. For the follow on task one can precompute relevant factorials and thus falling factorials for polynomial monomial derivatives' coefficients. One can compute $P(a)$ using the solution from the first part, followed by $O(n)$ additional multiplies, and then use finite differences to execute this task with an additional 0 multiplies and $O(n(b-a))$ additions.

[One ought to read books on mathematical computing and obtain As in courses like "Scientific Computing". The follow on task generalises a task I was given in "Probability Models" to compute the first 25 sums of the first n squares.]

Python Implementation:

```
def f(x, a):
    answer = x
    for b in range(len(a) - 1, 0, -1):
        answer *= a[b]
        answer += a[b - 1]
    print(answer)

av = [2 for a in range(100)]
av[0] = 1
for a in range(3):
    for b in range(1, 100):
        av[b] = av[b] + av[b - 1]
print(av)
```

`Write an algorithm to compute the first Bingo given a vector of pairs representing the` $n \times n$ `grid entries in the order they are called out.`

Simply store a vector of row, column, and diagonal sums thus far and iterate through, checking after each instance after n in the just updated sums for the completion condition of $== n$. Perhaps there exists a good slightly tighter version with comparisons and a storing of the max sum thus far.

```
# Given avv and n.
```

```
av = [0 for a in range(2 * n + 2)]
for a in range(n - 1):
    av[avv[a][0]] += 1
    av[n + avv[a][1]] += 1
    if avv[a][0] == avv[a][1]:
        av[2 * n] += 1
    if avv[a][0] == n - 1 - avv[a][1]:
        av[2 * n + 1] += 1
for a in range(n - 1, len(avv)):
    av[avv[a][0]] += 1
    if av[avv[a][0]] == n:
        print(a + 1)
        break
    av[n + avv[a][1]] += 1
    if av[n + avv[a][1]] == n:
        print(a + 1)
        break
    if avv[a][0] == avv[a][1]:
        av[2 * n] += 1
        if av[2 * n] == n:
            print(a + 1)
            break
    if avv[a][0] == n - 1 - avv[a][1]:
        av[2 * n + 1] += 1
        if av[2 * n + 1] == n:
            print(a + 1)
            break
```

`Write an algorithm to compute the square root of a number.`

Literature. One can compute the square root of a number using Fixed Point Iteration, also known as the Mechanic's Rule.

`Write an algorithm to compute the cosine of a number.`

Literature. PARI/GP, WolframAlpha, Mathematica or by Taylor's Theorem With Remainder one can compute an approximation.

`Write an algorithm to multiply integers.`

Literature. Harvey and van der Hoeven, Furer, Schonhage-Strassen, Karatsuba.

`Write an algorithm to compute compute` $n!$.

Literature. Swinging factorial and beyond. In a more general real world multiplication setting one can execute a multiplication algorithm or convolution in the processing order of smallest 2 vectors in our set, until we are left with the final answer. This may lead to a runtime asymptotic analysis of $\approx O(n \cdot (\log(n))^2)$. However, this is not such a setting as we have much more additional structure to exploit. For $n = 10^8$ one could use a sieve to produce the relevant primes in the factorisation. Then one could compute their exponents but amortised analysis is needed from hereon out for a proper clear, technical, precise writeup. Then one can execute the first steps of binary exponentiation. And finally, one can take the whole product multiplication in the optimal processing order.

`When can parallelism make one's algorithms run faster? When could it make one's algorithms run slower?`

When someone composes an effective algorithm to utilise it. Divided into sub tasks that can be executed independently of eachother without communication or shared resources. Some degree of sharing can be permitted as long as there is a speedup.

`Bobo the amoeba has a` 0.25`,` 0.25`, and` 0.5 `chance of producing` 0`,` 1`, or` 2 `offspring, respectively. Each of Bobo's descendants also have the same probabilities. What is the probability that Bobo's lineage dies out?`

Solving the natural resultant equation, the extinction probability of this stochastic branching process is the smallest non negative solution of $x = P(x) = \frac{1}{4} + \frac{1}{4} \cdot x + \frac{1}{2} \cdot x^2$ is $\boxed{\frac{1}{2}}$.

`How can one generate a random number between` 1`-`15 `with only` 1 `die?`

An optimal algorithm is to use the first roll to determine the number's remainder modulo 3. And then roll until the first non-6 determines which of the 5 buckets, partitions the number is in. Optimality follows from a decision tree depth argumentation. As $6^n \equiv 6 \pmod{15}$ one cannot guarantee halting because the number of strings of rolls [continued beyond the realised halt arbitarily as needed] corresponding with each outcome would need to be the same. And if more than $6^n - 6$ prefixes halt then a pigeonhole principle argumentation at this level produces a contradiction on the maximum already having probability $> \frac{1}{15}$. In our construction there are precisely minimally 6 sequences of rolls which remain alive.

`One has a` $50 - 50$ `mixture of` 2 `standard normal distributions. How far apart do the means need to be in order for this distribution to be bimodal?`

This distribution becomes bimodal when $(f+g)'$ has 3 zeroes moment i.e. when $(f+g)''$ crosses at their inflection points i.e. when their means are precisely 2 standard deviations $\boxed{2}$ apart.

`Given draws from a normal distribution with known parameters, how can one simulate draws from a uniform distribution?`

A probability density function lookup correspondence map.

`Some couples decide to have children until their first girl, after which they stop having children. What is the expected gender ratio of the children that are born? What is the expected number of children each couple will have?`

Linearity of expectation on each instance of a child being born the ratio is $\frac{1-P}{P} = \boxed{1}$ and thus the expected number of children is $1 + \frac{1-P}{P} = \frac{1}{P} = \boxed{2}$.

`How many ways can one split` 12 `people into` 3 `teams of` 4`?`

$\binom{12}{4,4,4}\frac{1}{3!} = \boxed{5775}$. See Twelvefold Way Richard Stanley and Putnam Notes.

`One's hash function assigns each object to a number between 1-10, each with equal probability. With` 10 `objects, what is the probability of a hash collision? What is the expected number of hash collisions? What is the expected number of hashes that are unused?`

$$1 - \frac{10!}{10^{10}} = \boxed{\frac{1561933}{1562500}} \approx 0.99637$$

Linearity of expectation $\frac{\binom{10}{2}}{10} = \boxed{\frac{9}{2}}$

$$10\left(1 - \left(\frac{9}{10}\right)^{10}\right) = \boxed{6.513215599}$$

`What's the difference between a MAP, MOM, MLE estimator? In which` cases would one want to use each?

MAP: Maximum A Posteriori Estimate is the point which maximises the posterior distribution. Seen as a regularisation of Maximum Likelihood Estimation. Rather than computing the point which maximises the likelihood function, which is equivalent with MAP from a uniform prior.

MOM: Method Of Moments, of Chebyshev initially, expresses population moments, expected values of powers, as functions of the parameters of interest. Set them equations to sample moments. Solutions are estimates for parameters. Consistent, often biased, yadda yadda.

MLE: Maximum Likelihood Estimate aforementioned.

What is a confidence interval and how does one interpret it?

In frequentist and Bayesian statistics a confidence/credence interval can refer to the probability that some observation occurs under the model for producing the data or it can mean an interval inside which mass lies technically an integral of a probability density function.

What is unbiasedness as a property of an estimator?

The expected value of the estimator is the true underlying population value e.g. $\mathrm{E}[\hat{z} - z] = 0$. A classic example is of German tanks in World War II. If one observes distinct numbers on all n tanks with maximum value m then the Maximum Likelihood Estimator for the number of tanks may be m but the Minimum Variance Unbiased Estimator MVUE is $m\left(\frac{n+1}{n}\right) - 1$.

What is the Curse Of Dimensionality?

This usually refers to the fact that in high dimensional spaces more points are prone to being near the boundary of the convex hull of the input point set. So models may be weaker and techniques such as Support Vector Machines, which are performant on such datasets, ought to be used.

How does one deal with some predictors being missing?

Literature. There exist tons of different techniques for imputation. Can be mean, median, algorithms for generating plausible values there prior to executing Machine Learning algorithms upon the dataset or one can consider, depending on risk analysis, upsides and downsides of errors, computational reasons, ignoring that predictor entirely.

`What is the main idea behind ensemble learning? If I had many different models that predicted the same response variable, what might I want to do to incorporate all of the models? Would one expect this to perform better than an individual model or worse?`

The key idea has to do with mixing up a few models, can be based on different subsets, and produce less error on a test set. Random forests. Can take mean or vote of prediction of the individual trees. It can perform better under certain metrics of performance. See Elements Of Statistical Learning.

`One has 100 mathletes and 100 math problems. Each mathlete gets to choose 10 problems to solve. Given data on who got what problem correct, how would one rank the problems in terms of difficulty?`

Literature, variants on Rasch Model, etc. lowkey like that Google Code Jam 2021 Qualifications Round Task 5 Cheating Detection where I aced the threshold parameter and nearly everything.

`One has 5000 people that rank 10 sushi in terms of saltiness. How would one aggregate this data to estimate the true saltiness rank in each sushi?`

This is an ordering, not a quantitative input. So we might consider a median first mean approach for each sushi. Other domains like this include aggregation of polls to rank college American Football teams and those have financial implications and so one assumes there exists a vast literature on this topic anyways from the economics and mathematics permutations side of things.

[Insofar as saltiness is a perception, perhaps joke about writing a model of the physics of the salt receptors down to the micro particle scale and then using quantum entanglement to ping the execution engine with 0 latency arbitrage. Or run a sodium analysis like they would to produce a label on a product under United States Of America law depending on what it really is that one is after. Vague task statements can be clarified with interviewers, who can choose to let one run with one's own stated assumptions and clarifications. In terms of demonstrating a comprehension of reality, incentives, rules, it is a good idea to bring up United States Of America laws.]

`Ideate upon an algorithm to detect plagiarism in online content.`

Well I suppose we need to write code so even if they make a typo on quotation marks we still somehow look at gaps and block out quotations. And then on the rest of the text we can execute some Google search type text strings algorithms. And perhaps we can plagiarise some stuff and see if our model is doing a good or bad job on some examples of plagiarism. Perhaps I misunderstand and in fact this is a very exceedingly tricky task and I somewhat feel bad for uncertain course instructors who are presented with a single plagiarism number on a 0-100 scale.

One runs a restaurant and is approached by Groupon to run a deal. What data would one ask from them in order to determine whether or not to do the deal?

It is business of profit pursuing entities, so depending on laws and a general sense for trust, one might contemplate then write the precise complete data one wants from from the Groupon firm. Run a historical analysis related to firms similar in some meaningful ways to one's own. Avoid being statistically conned by shady dark arts from the Groupon firm. Find the public written record of a comprehensive set of firms the Groupon firm worked with in the past, including firms where they failed i.e. they ran a couple coupons with and then stopped cooperating. One wants something complete, if one names a variable and Groupon produces a cherry picked set of firms Groupon worked with to suggest something, that is a red flag.

One is tasked with improving the efficiency of a subway system. Where would one start?

Literature. "Braess's Paradox", that adding a road can slow down drive times and decrease network throughput, and other efficiency notions tasks. Evaluation metrics matter as well as the downstream effects of a local policy deviation, and further potential paths for future system growth.

Dialogues

I read the firm's website. Cool blog. There are many ways I should have written more performant code on the coding round. I could have used... [a performant assembly Fibonacci Euler Ilic Splay Tree Variant] I realised that for the... task you all would probably prefer something like C++:

I recently learned more about optimal mathematical computing from my Scientific Computing course textbook. I have yet to be admitted to the University Of Texas At Austin Master Of Science In Computer Science program, but I spoke with the graduate advisor, and expect to gain admission in the next 4 or 5 weeks.

I will consider posting concrete examples from my own interviews that were not covered by a signed Non Disclosure Agreement here at some point in time.

Firms [How firm Is The Firm? Under Which Metric?]:

Open Artificial Intelligence, XTX Markets, Renaissance Technologies, Jump Trading, Hudson River Trading, Jane Street Capital, Citadel Securities, D.E. Shaw, DRW Trading Group, Headlands Technologies, Five Rings Trading, Susquehanna International Group, Optiver, IMC Financial Markets, Tower Research Capital, Two Sigma, Akuna Capital, Virtu Financial, Ansatz Capital

3Red Partners, Allston Trading, Alphabit Trading, Appaloosa Management, AQR Trading, Aquatic Trading, Arbor Ventures, Arrowgrass Capital Management, Aspect Capital, Avatar Securities, Axiom Markets, Balyasny Europe, Baupost Group, Belvedere Trading, Blackedge Trading, BlackRock, BlueBay Asset Management, BlueFin Trading, Blue Mountain Capital, Bridgewater Associates, Budo Trading, Capula Invesment, Caxton Europe, Chicago Trading Company, Coatue Management, CQS LLP, Crabel Capital Management, Davidson Kempner Capital Management, Domeyard Trading, DV Trading, Eagle Seven Trading, Edgehog Trading, Eisler Capital, Elliott Asset Management, Epoch Capital, Eschaton Trading, Final Trading, First New York, First Quadrant, Flow Traders, Gelber Group, Geneva Trading, Grace Hall Trading, Group One Trading, GTS, HAP Capital, Hehmeyer Trading, HNK Alpha, Hold Brothers, Istra Research, Kalshi, Lone Pine Capital, Maize Capital, Marquette Partners, Maven Securities, Millenium Management, Nova Satus Trading, Och-Ziff Management, Odey Asset Management, Old Mission Capital, Peak 6 Investments, Pershing Square Capital Management, PNT Financial, Point 72 Trading, Process Driven Trading, Quantlab, Quora, Radix Trading, RSJ Algorithmic Trading, Schonfeld Group, Seven Points Capital, Simplex Trading, Source Capital, Squarepoint Capital, STX Group, Tenzan Capital, Tiger Global Management, TGS Trading, Tradebot Systems, Tradelink Trading, TT International Investment Management, Tudor Capital, Vatic Labs, Vivienne Court Trading, Voleon Trading, Weiss Asset Management, WH Trading, Winton Capital Management, Wolverine Trading, XR Trading

[LinkedIn, Public Written Records, Y Combinator, a16z, Huxley, GQR, Kaizen Finance, EKA Finance, University Of Texas At Austin Lists Trawling]

Thanks

I deeply thank the people who taught me and will continue to teach others, young and old, for years to come.

Hans Magnus Enzensberger, George Lenchner, Sam Baethge, Max Warshauer, Jian Shen, David Patrick, Richard Rusczyk, Mathew Crawford, Sandor Lehoczky, Palmer Mebane, Naoki Sato, Valentin Vornicu, Titu Andreescu, Zuming Feng, Po-Shen Loh, Yufei Zhao, Cosmin Pohoata, Pranav Sriram, Evan Chen, Nets Katz, Kiran Kedlaya, Joe Gallian, David Rusin, Inna Zakharevich, Peter Winkler, Alexander Bogomolny, Antti Laaksonen, Colin Hughes, Jeff Erickson, Umesh Vazirani, Robert Tarjan, Donald Knuth, Ronald Graham, Richard Stanley, Zach Wissner-Gross, Oliver Roeder, Ken Ono, Pradeep Mutalik, Gadi Aleksandrowicz, Oded Margalit, James Shearer, Don Coppersmith, Mark Joshi, Timothy Falcon Crack, Paul Wilmott, Xinfeng Zhou, Frederick Mosteller, Dan Stefanica, Marcos Lopez De Prado, Geoffrey Grimmett, David Stirzaker, Gordan Zitkovic, Milica Cudina, Dusan Djukic, Fedor Petrov, Geoff Smith, C J Bradley, and all other composers of tasks, textbooks, puzzles, and source notes.